KATHOLIEKE UNIVERSITEIT LEUVEN

The Joint Block Diagonalization (JBD) problem: *a tensor framework.*

Dimitri Nion & Lieven De Lathauwer

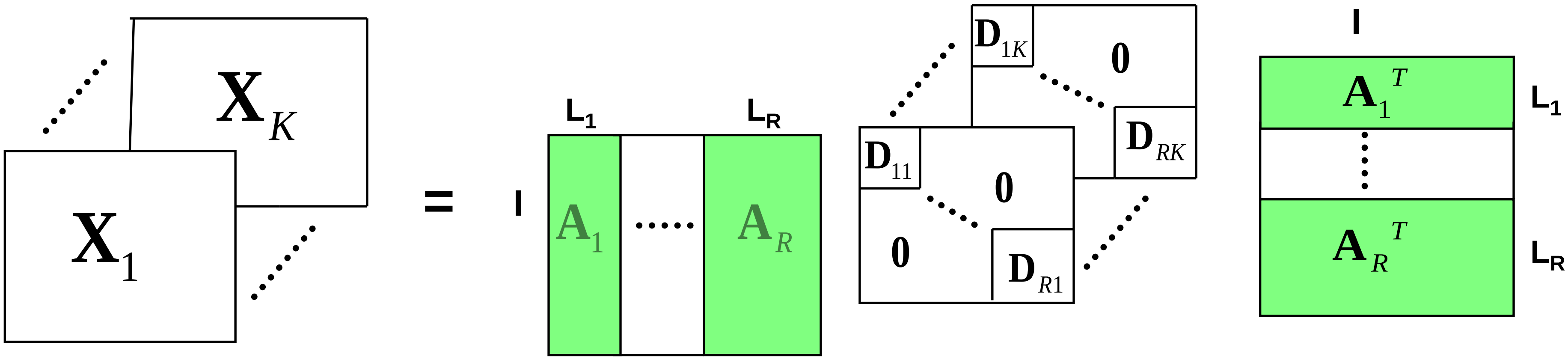
K.U. Leuven, Kortrijk campus, Belgium

E-mails: Dimitri.Nion@kuleuven-kortrijk.be

Lieven.DeLathauwer@kuleuven-kortrijk.be

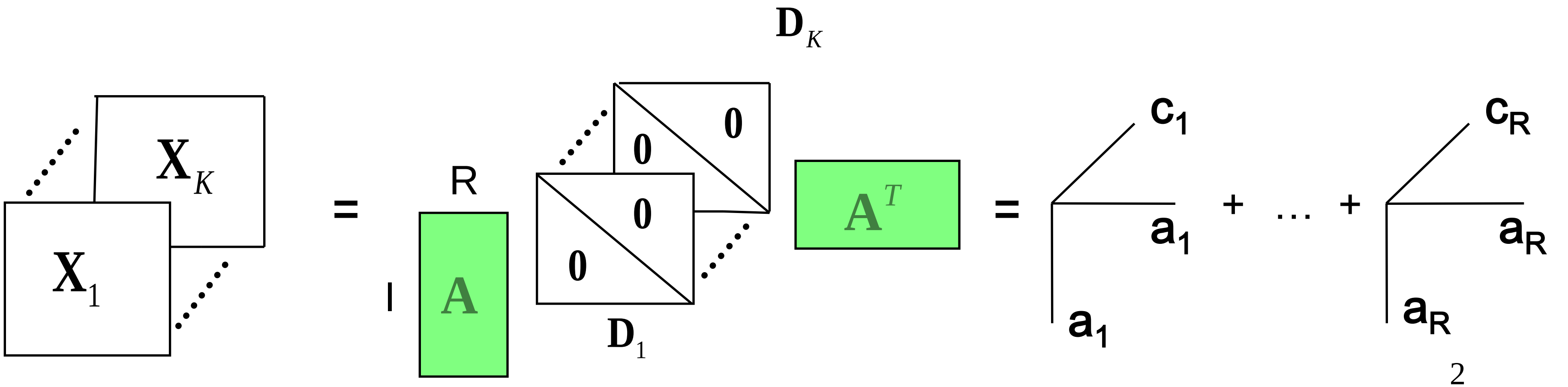
TDA 2010, Monopoli, Italy, September 13-17, 2010

Joint-Block-Diagonalization (JBD): model

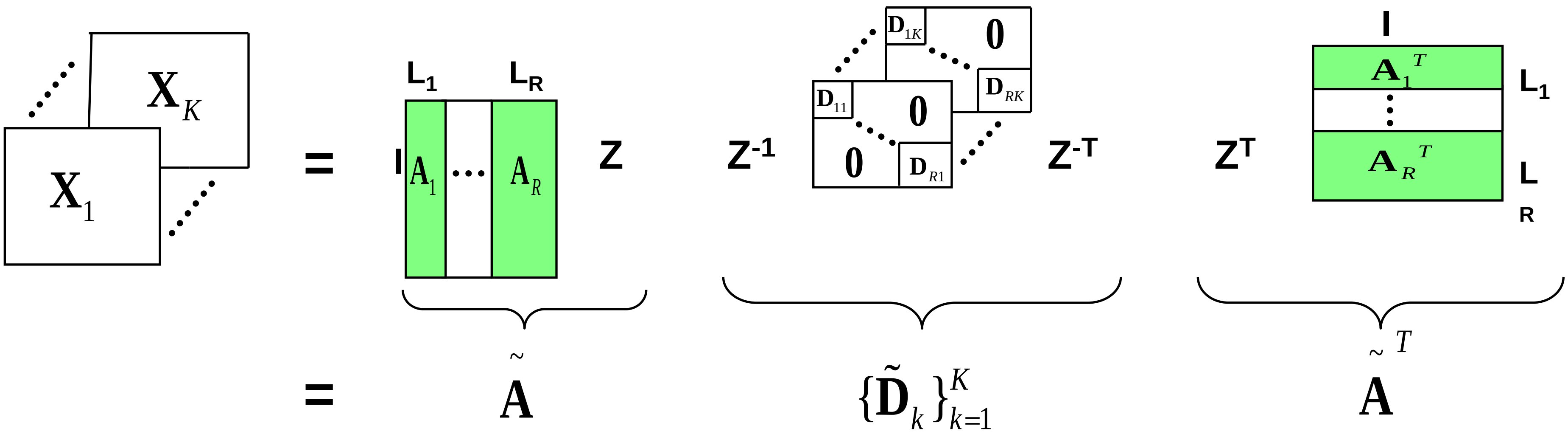


$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T \ (+\mathbf{N}_k) \quad or \quad \mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^H \ (+\mathbf{N}_k), \ k = 1,\ldots, K$$

JBD is a generalization of JD (Joint Diagonalization)/INDSCAL



JBD : ambiguities



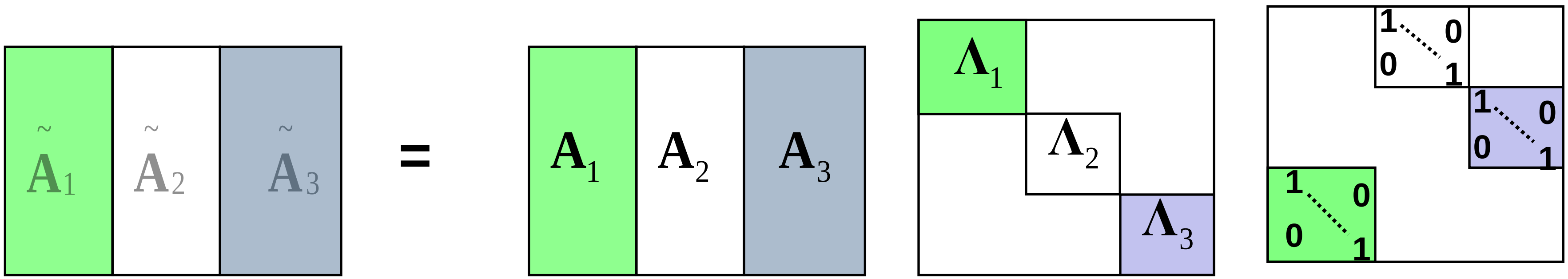
Observation: if you choose **Z** arbitrary, you lose the JBD structure.

Question: what is the structure of **Z** such that the JBD model is still valid?

JBD : essential uniqueness

The JBD of $\{\mathbf{X}_k\}_{k=1}^{K}$ is said essentially unique if $\mathbf{Z} = \boldsymbol{\Lambda}\boldsymbol{\Pi}$

- $\boldsymbol{\Lambda}$ an arbitrary block-diagonal matrix,
- $\boldsymbol{\Pi}$ an arbitrary block-wise permutation matrix.



Solving a JBD problem

⟺ Estimation of $\{\mathrm{Span}(\mathbf{A}_r)\}_{r=1,\ldots,R}$ in an arbitrary order

JBD: State of the art

JBD is becoming popular signal processing tool in applications such as:

- Blind Source Separation (BSS) of convolutive mixtures in the time-domain,
- Independent Subspace Analysis.

Two approaches in the literature:

- **Approach 1:** Unitary-JBD *[Abed Meraim and Belouchrani, 2004]*

« **A** is a square unitary matrix » ($\mathbf{A}^T \mathbf{A} = \mathbf{I}$)

- **Approach 2:** Non-Unitary JBD

▪ Approach 2.1: *[H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]*

« **A** is tall and full column-rank. »

→ Indeed, their approach only works if **A** is a square non-unitary matrix.

▪ Approach 2.2: *This talk*

« **A** can be a tall, square or fat non-unitary matrix »

→ JBD is a particular instance of Block-Component-Decompositions

→ Computation by a gradient-based algorithm

→ In the square case, better performance than 2.1

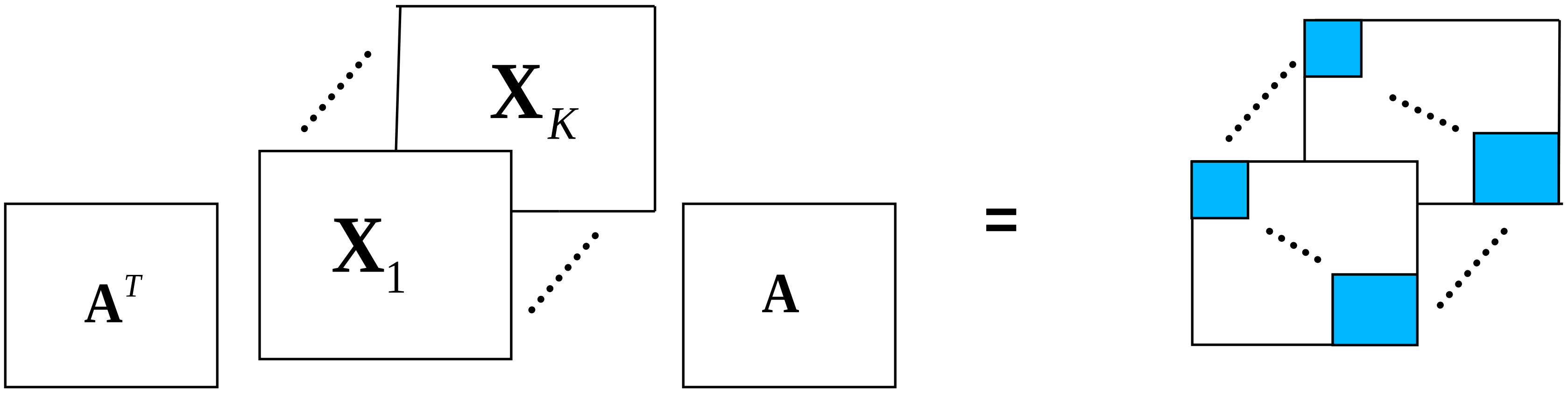
Joint-Block-Diagonalization : state of the art (1)

- Approach 1: Unitary-JBD [Abed Meraim and Belouchrani, 2004]

➢ **A** is square unitary matrix ($\mathbf{A}^T\mathbf{A} = \mathbf{I}$)

$$\mathbf{X}_k = \mathbf{A}\,\mathbf{D}_k\,\mathbf{A}^T\ (+\ \mathbf{N}_k) \iff \mathbf{A}^T\,\mathbf{X}_k\,\mathbf{A} = \mathbf{D}_k +\ (\mathbf{A}\,\mathbf{N}_k\,\mathbf{A}^T)$$

$$\max_{\mathbf{A}} \sum_{k=1}^{K} \left\| bdiag(\mathbf{A}^T \mathbf{X}_k \mathbf{A}) \right\|_F^2 \quad or \quad \min_{\mathbf{A}} \sum_{k=1}^{K} \left\| offbdiag(\mathbf{A}^T \mathbf{X}_k \mathbf{A}) \right\|_F^2$$



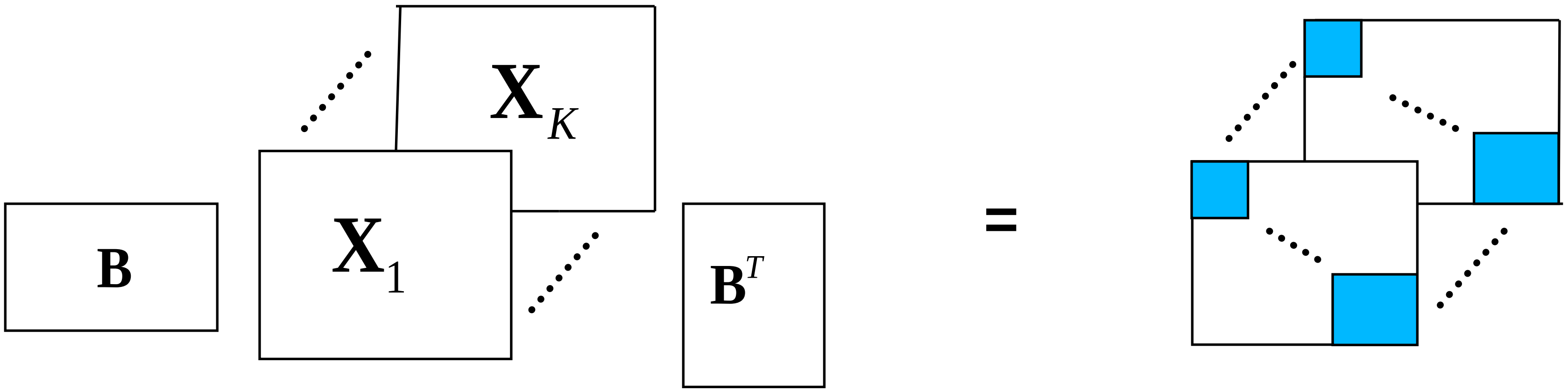
Joint-Block-Diagonalization : state of the art (2)

- Approach 2: Non-Unitary-JBD [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]

➢ **A** is tall and full column-rank (Let $\mathbf{B} = \mathbf{A}^{\dagger}$, then $\mathbf{BA} = \mathbf{I}$)

$$\mathbf{X}_k = \mathbf{A}\,\mathbf{D}_k\,\mathbf{A}^T \,(+\, \mathbf{N}_k) \iff \mathbf{B}\mathbf{X}_k\mathbf{B}^T = \mathbf{D}_k + (\mathbf{B}\,\mathbf{N}_k\,\mathbf{B}^T)$$

$$\max_{\mathbf{B}} \sum_{k=1}^{K} \left\| bdiag(\mathbf{B}\mathbf{X}_k\mathbf{B}^T) \right\|_F^2 \quad or \quad \min_{\mathbf{B}} \sum_{k=1}^{K} \left\| offbdiag(\mathbf{B}\mathbf{X}_k\mathbf{B}^T) \right\|_F^2$$



Joint-Block-Diagonalization : state of the art (3)

- Approach 2: Non-Unitary-JBD [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]

➢ 2 gradient-descent based algorithms, $\mathrm{JBD}_{\mathrm{OG}}$ and $\mathrm{JBD}_{\mathrm{ORG}}$ to solve

$$\min_{\mathbf{B}} \phi_{\text{off}} = \sum_{k=1}^{K} \left\| offbdiag(\mathbf{B}\mathbf{X}_k \mathbf{B}^T) \right\|_F^2 \qquad (1)$$

- Drawbacks of approach 2:

➢ **B**=0 is a trivial minimizer

➢ Under-determined case (**A** fat, I<N) not handled, since it is assumed that **BA**=**I**

➢ Indeed, the over-determined case (**A** tall, I>N) is not successfully handled either because if $\mathbf{B} = [\mathbf{B}_1^T, \mathbf{B}_2^T]^T = \mathbf{A}^\dagger$ is solution of an exact JBD problem, i.e.,

$$offbdiag(\mathbf{B}\mathbf{X}_k\mathbf{B}^T) = offbdiag(\mathbf{B}\mathbf{A}\mathbf{D}_k\mathbf{A}^T\mathbf{B}^T) = offbdiag(\mathbf{D}_k) = 0, \ \forall k$$

then $\tilde{\mathbf{B}} = [(\mathbf{A}^\perp)^T, \mathbf{B}_2^T]^T$ is also solution of (1) but not solution of the JBD problem

because $\tilde{\mathbf{B}}\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ X & X \end{bmatrix}$ is not full rank. Idem for $\tilde{\mathbf{B}} = [\mathbf{B}_1^T + (\mathbf{A}^\perp)^T, \mathbf{B}_2^T]^T$

Joint-Block-Diagonalization : our contributions

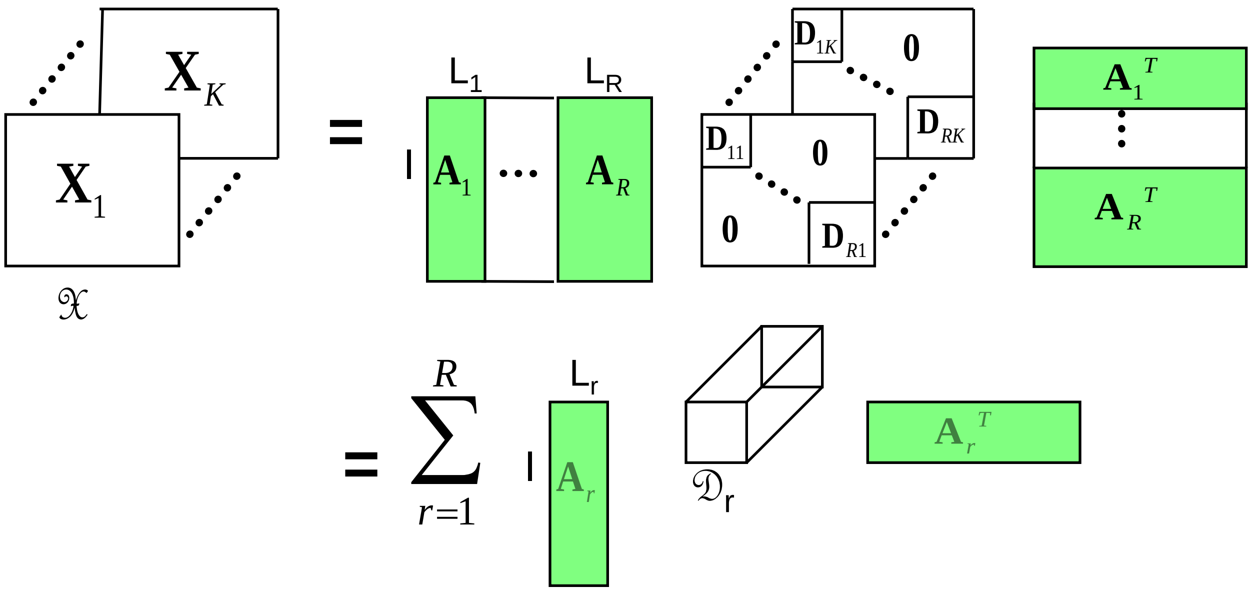
- **Starting point:** the gradient-descent based algorithms JBD_{OG} and JBD_{ORG} of [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010] for Non-Unitary JBD can only handle the case where **A** is square (I=N).

- **Main motivation:** Propose a novel technique to solve non-unitary JBD problems that can handle exactly-, over- and under- determined cases (i.e., **A** may be square, tall or fat).

- **Main contributions:**

  - Formulate JBD as a tensor decomposition fitting problem;
  - Build a Conjugate Gradient (CG) algorithm to compute the tensor decomposition;
  - In the over-determined case, build a good starting point for any JBD algorithm;
  - Application: blind source separation via Independent Subspace Analysis (ISA).

JBD in tensor format : link to BCD



$$\mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r$$

(or $\mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r^*$

in case of hermitian symmetry)

JD ⟺ particular case of candecomp-parafac

JBD ⟺ particular case of BCD-(L_r, M_r, .), (« Block-Component-Decomposition in rank-(L_r, M_r, .) terms »)

JBD in tensor format : conditions for essential uniqueness

- BCD-(L_r , M_r , .) : $$\mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{B}_r$$ where **A**=[$\mathbf{A}_1$,…,$\mathbf{A}_R$] is I by N
B=[$\mathbf{B}_1$,…,$\mathbf{B}_R$] is J by Q

- Theorem [De Lathauwer, 2008]: Suppose that rank(**A**)=N, rank(**B**)=Q, K>2 and that the tensors $\{\mathcal{D}_r\}_{r=1,\ldots,R}$ are generic, then the BCD-(L_r , M_r , .) of $\mathcal{X}$ is essentially unique (Sufficient condition).

- JBD : $$\mathcal{X} = \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r$$ where **A**=[$\mathbf{A}_1$,…,$\mathbf{A}_R$] is I by N

- The same theorem can be invoked (the proof still holds with **A** instead of **B**)
- In summary, it means that JBD is generically unique if

K>2 and rank(**A**)=N

- This is only a sufficient condition: uniqueness still holds but is harder to prove in several cases where the condition is not satisfied.
- For instance, uniqueness may still hold when rank(**A**)=I (**A** fat, I<N)

JBD in tensor format : cost function

$$\min_{\{\mathcal{D}_r\}_{r=1}^{R},\mathbf{A}} \phi_{LS} = \|\mathcal{N}\|_F^2 \qquad \textit{with} \quad \mathcal{N} = \mathcal{X} - \sum_{r=1}^{R} \mathcal{D}_r \bullet_1 \mathbf{A}_r \bullet_2 \mathbf{A}_r$$

$$= \sum_{k=1}^{K} \|\mathbf{N}_k\|_F^2 \qquad \textit{with} \quad \mathbf{N}_k = \mathbf{X}_k - \sum_{r=1}^{R} \mathbf{A}_r \mathbf{D}_{kr} \mathbf{A}_r^T$$

$$= \|\mathbf{N}\|_F^2 \qquad \textit{with} \quad \mathbf{N} = \mathbf{X} - (\mathbf{A} \odot \mathbf{A})\mathbf{D}$$

Where:

- **A** $\odot$ **A** = [$\mathbf{A}_1 \otimes \mathbf{A}_1$, …, $\mathbf{A}_R \otimes \mathbf{A}_R$] is the Khatri-Rao product (block-wise Kronecker product)
- **X** and **N** are the I^2xK matrix unfoldings of the IxIxK tensors $\mathcal{X}$ and $\mathcal{N}$, resp.

JBD in tensor format : derivation of the gradient

Real-valued data

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T + \mathbf{N}_k$$

$$\begin{cases} \nabla_{\mathbf{A}}(\phi_{LS}) = -2\sum_{k=1}^{K}(\mathbf{N}_k^T\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^T), \\ \nabla_{\mathbf{D}}(\phi_{LS}) = -2(\mathbf{A}\odot\mathbf{A}^T)\mathbf{N} \end{cases}$$

Complex-valued data, standard symmetry

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^T + \mathbf{N}_k$$

$$\begin{cases} \nabla_{\mathbf{A}^*}(\phi_{LS}) = -\sum_{k=1}^{K}(\mathbf{N}_k^T\mathbf{A}^*\mathbf{D}_k^* + \mathbf{N}_k\mathbf{A}^*\mathbf{D}_k^H), \\ \nabla_{\mathbf{D}^*}(\phi_{LS}) = -(\mathbf{A}\odot\mathbf{A})^H\mathbf{N} \end{cases}$$

Complex-valued data, hermitian symmetry

$$\mathbf{X}_k = \mathbf{A}\mathbf{D}_k\mathbf{A}^H + \mathbf{N}_k$$

$$\begin{cases} \nabla_{\mathbf{A}^*}(\phi_{LS}) = -\sum_{k=1}^{K}(\mathbf{N}_k^H\mathbf{A}\mathbf{D}_k + \mathbf{N}_k\mathbf{A}\mathbf{D}_k^H), \\ \nabla_{\mathbf{D}^*}(\phi_{LS}) = -(\mathbf{A}\odot\mathbf{A}^*)^T\mathbf{N} \end{cases}$$

Conjugate Gradient algorithm for JBD: JBD-CG

while (stop criterion not satisfied)

1- Compute $\nabla_{\mathbf{A}}$ and $\nabla_{\mathbf{D}}$ from $\mathbf{D}$ and $\mathbf{A}$

2 - Update search directions

2.1- Compute β (e.g. with stabilized Polak - Ribière formula)

2.2 - New search directions :

$$\mathbf{d}_{\mathbf{A}} \leftarrow -\nabla_{\mathbf{A}} + \beta\, \mathbf{d}_{\mathbf{A}}$$

$$\mathbf{d}_{\mathbf{D}} \leftarrow -\nabla_{\mathbf{D}} + \beta\, \mathbf{d}_{\mathbf{D}}$$

3 - Joint Exact Line Search :

$$(\alpha_{\mathbf{A}}, \alpha_{\mathbf{D}}) \leftarrow \arg\min_{\alpha_{\mathbf{A}}, \alpha_{\mathbf{D}}} \phi_{LS}(\mathbf{A} + \alpha_{\mathbf{A}} \mathbf{d}_{\mathbf{A}}, \mathbf{D} + \alpha_{\mathbf{D}} \mathbf{d}_{\mathbf{D}})$$

4 - Update unknwons :

$$\mathbf{A} \leftarrow \mathbf{A} + \alpha_{\mathbf{A}} \mathbf{d}_{\mathbf{A}}$$

$$\mathbf{D} \leftarrow \mathbf{D} + \alpha_{\mathbf{D}} \mathbf{d}_{\mathbf{D}}$$

End

Step 3 of JBD-CG : Joint Exact Line Search

$$\min_{\alpha_{\mathbf{A}},\alpha_{\mathbf{D}}} \phi_{LS}(\mathbf{A}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}, \mathbf{D}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}})$$

$$\phi_{LS}(\mathbf{A}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}, \mathbf{D}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}}) = \left\| \left[(\mathbf{A}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}}) \odot (\mathbf{A}+\alpha_{\mathbf{A}}\mathbf{d}_{\mathbf{A}})\right](\mathbf{D}+\alpha_{\mathbf{D}}\mathbf{d}_{\mathbf{D}}) - \mathbf{X} \right\|_F^2$$

$$= \alpha_{\mathbf{D}}^{\ 2} Q_2(\alpha_{\mathbf{A}}) + 2\alpha_{\mathbf{D}} Q_1(\alpha_{\mathbf{A}}) + Q_0(\alpha_{\mathbf{A}}) \qquad (1)$$

→ Solve $\dfrac{\partial \phi_{LS}(\alpha_{\mathbf{A}},\alpha_{\mathbf{D}})}{\partial \alpha_{\mathbf{D}}} = 0$ ⇨ $\alpha_{\mathbf{D}} = \dfrac{-Q_1(\alpha_{\mathbf{A}})}{Q_2(\alpha_{\mathbf{A}})} \quad (2)$

→ Substitute (2) in (1): $\phi_{LS}(\alpha_{\mathbf{A}}) = \dfrac{-Q_1^{\ 2}(\alpha_{\mathbf{A}}) + Q_0(\alpha_{\mathbf{A}})Q_2(\alpha_{\mathbf{A}})}{Q_2(\alpha_{\mathbf{A}})} \quad (3)$

→ Minimize (3) w.r.t. $\alpha_{\mathbf{A}}$ (degree 7 polynomial rooting)

→ Given $\alpha_{\mathbf{A}}$, $\alpha_{\mathbf{D}}$ is given by (2)

JBD in tensor format : a closed form solution (when A is full column-rank)

- Idea: when **A** is square or tall (full column-rank), the exact (noise-free) JBD problem can be solved by eigenvalue analysis. In presence of noise, this solution can be used to initialize iterative algorithms.

- Derivation in the square case (I=N): Take two matrices $\left\{ \begin{array}{l} \mathbf{X}_1 = \mathbf{A}\mathbf{D}_1\mathbf{A}^T \\ \mathbf{X}_2 = \mathbf{A}\mathbf{D}_2\mathbf{A}^T \end{array} \right\}$

- We have $\mathbf{X}_{12} = \mathbf{X}_1\mathbf{X}_2^{-1} = \mathbf{A}\mathbf{D}_{12}\mathbf{A}^{-1}$ where $\mathbf{D}_{12}$ is block-diagonal.
- So $\mathbf{X}_{12}\mathbf{A}_r = \mathbf{A}_r[\mathbf{D}_{12}]_{rr}$,i.e., $Span(\mathbf{A}_r),\ \forall r,$ is an invariant subspace of $\mathbf{X}_{12}$.
- EVD: $\mathbf{X}_{12} = \mathbf{E}\Lambda\mathbf{E}^{-1}$, where Λ is diagonal.
- We have $\mathbf{A} = \mathbf{E}\mathbf{\Pi}$, where $\mathbf{\Pi}$ is a permutation matrix that groups the eigenvectors L_r by L_r.
- Find $\mathbf{\Pi}$ by checking the permuted block-diagonal structure of the matrices $\mathbf{E}^{-1}\mathbf{X}_k\mathbf{E}^{-T} = \mathbf{\Pi}\mathbf{D}_k\mathbf{\Pi}^T,\ k = 1,\dots,K$

Performance index

- We have : $\hat{\mathbf{A}} = \mathbf{A}\,\boldsymbol{\Lambda}\,\boldsymbol{\Pi} + \mathbf{N}$ where
  - $\boldsymbol{\Lambda}$ is an unknown block-diagonal matrix,
  - $\boldsymbol{\Pi}$ is an unknown block-wise permutation matrix,
  - $\mathbf{N}$ is the estimation error.

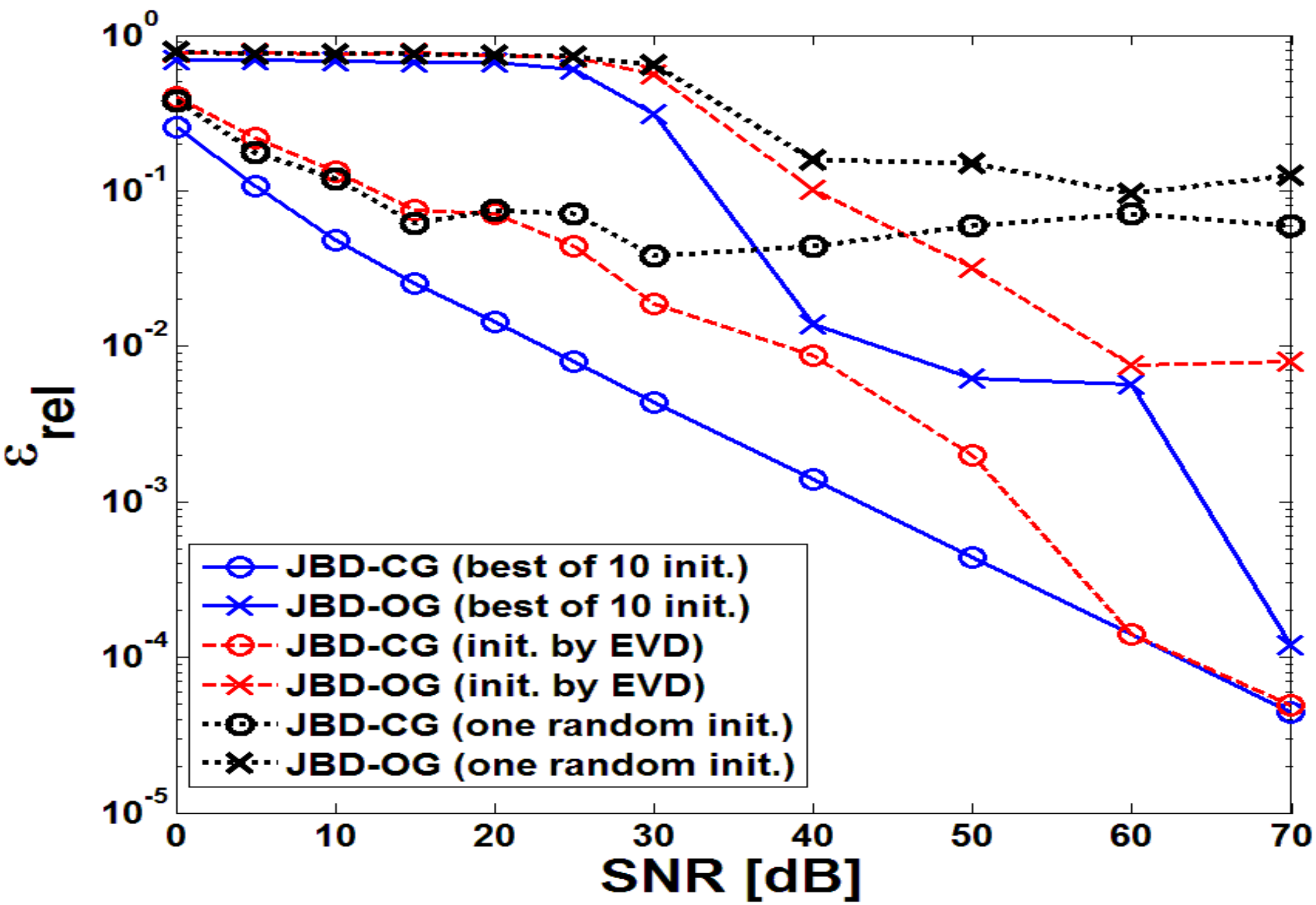
- Compute $\boldsymbol{\Pi}$ and $\boldsymbol{\Lambda}$ such that $\mathbf{A}\,\boldsymbol{\Lambda}\,\boldsymbol{\Pi}$ matches $\hat{\mathbf{A}}$ in the least squares sense.
- Compute the relative error:

$$\varepsilon_{\text{rel}} = \frac{\left\| \hat{\mathbf{A}} - \mathbf{A}\,\boldsymbol{\Lambda}\,\boldsymbol{\Pi} \right\|_F}{\left\| \mathbf{A} \right\|_F}$$

Numerical experiments (1)

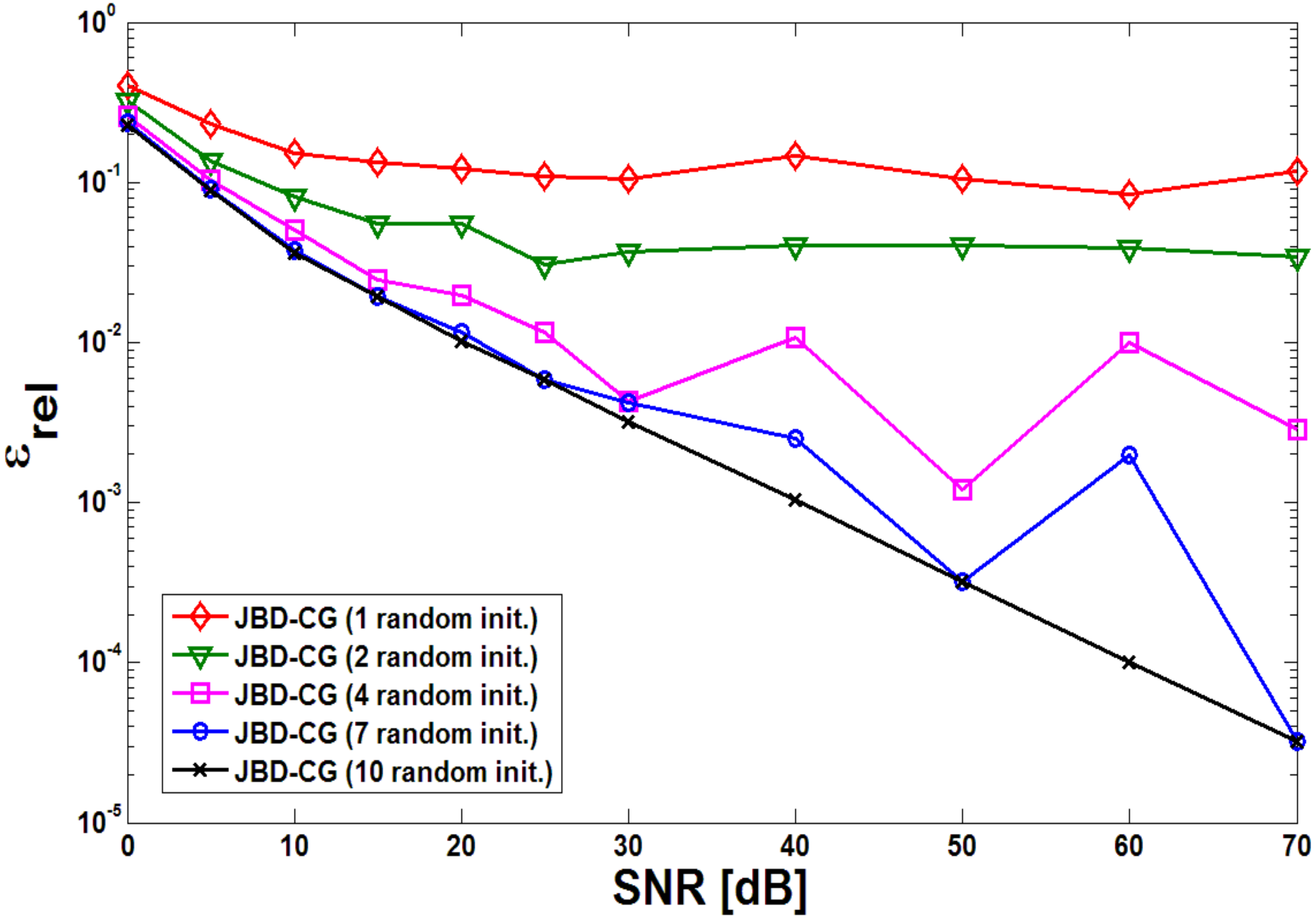
Exactly determined case: $I=N=6$, $L_1=L_2=L_3=2$, $R=3$, $K=30$

Comparison of our JBD-CG technique with JBD-OG of [H. Ghennioui, N. Thirion-Moreau, E. Moreau, 2008, 2010]

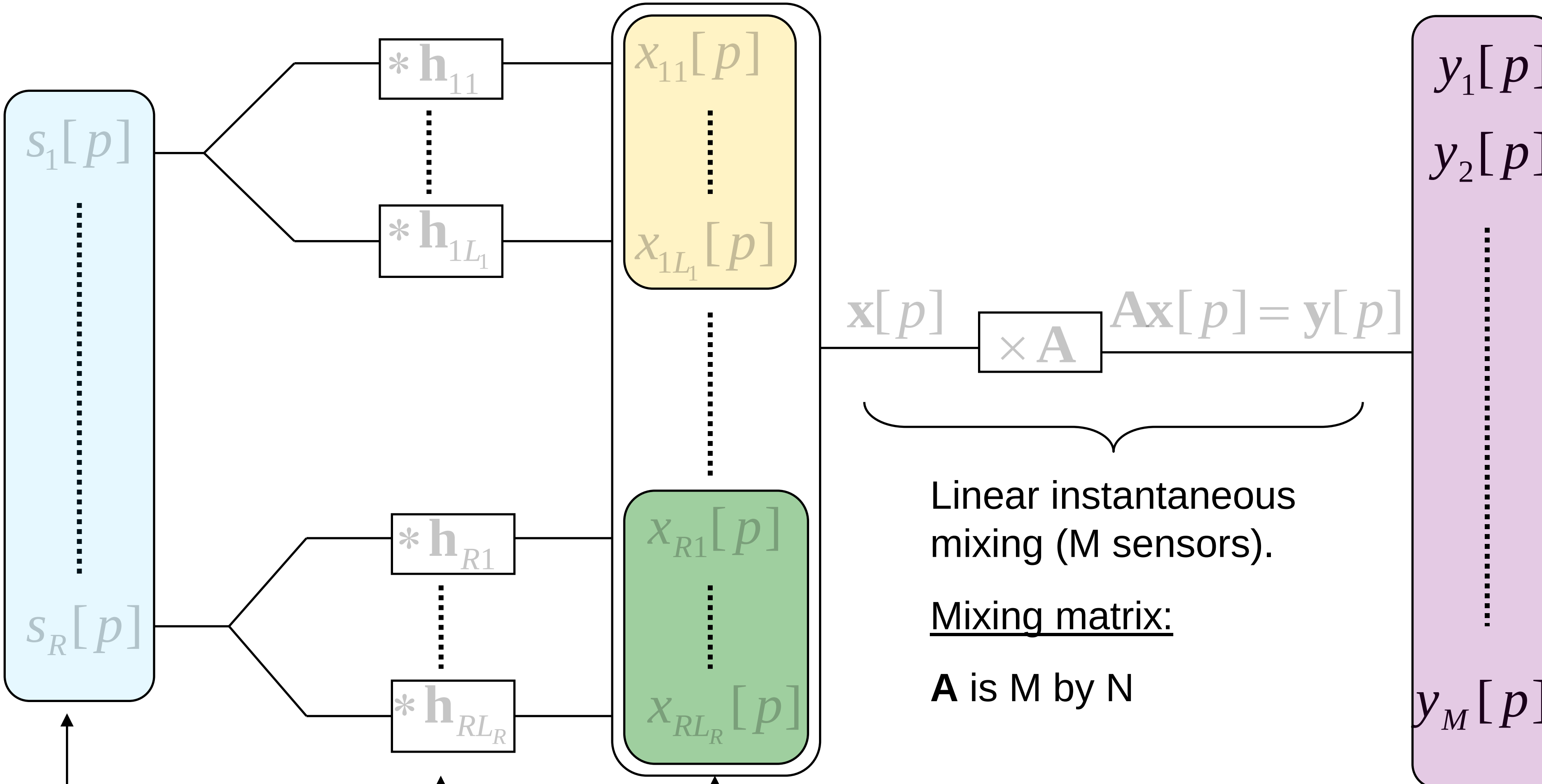


Numerical experiments (2)

Under determined case: $I=6$, $N=8$, $L_1=L_2=L_3=L_4=2$, $R=4$, $K=100$



Application: Blind Subspace Separation



Linear instantaneous mixing (M sensors).

<u>Mixing matrix:</u>

A is M by N

R **primary sources**

(mutually uncorrelated)

Set of FIR filters $N = \sum_{r=1}^{R} L_r$

N **Secondary sources:**

- Mutually correlated within the same subset
- Mutually uncorrelated if not in the same subset

Blind Subspace Separation via Second-Order-Statistics (SOS)

- $\mathbf{y}[p] = \mathbf{A}\mathbf{x}[p]$
- Covariance matrix:

$$\mathbf{R}_{\mathbf{yy}}[\tau] = E\{\mathbf{y}[p]\mathbf{y}^H[p+\tau]\} = \mathbf{A}\,E\{\mathbf{x}[p]\mathbf{x}^H[p+\tau]\}\mathbf{A}^H = \mathbf{A}\mathbf{D}\mathbf{A}^H$$

where **D** is block diagonal because on assumptions on secondary sources.

- Set of K covariance matrices:

$$\left\{\begin{array}{c} \mathbf{R}_{\mathbf{yy}}[\tau_1] = \mathbf{A}\mathbf{D}_1\mathbf{A}^H \\ \vdots \\ \mathbf{R}_{\mathbf{yy}}[\tau_K] = \mathbf{A}\mathbf{D}_K\mathbf{A}^H \end{array}\right\} \Rightarrow$$

JBD problem !

- Estimate **A** by JBD-ACG algorithm
- In the (over)-determined case (**A** square or tall), LS estimate of secondary sources:

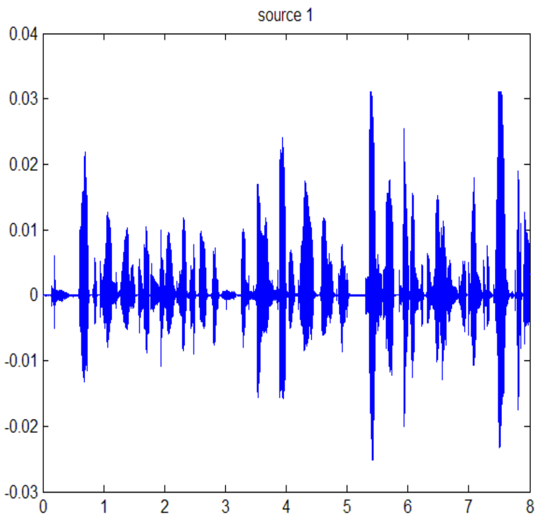
$$\mathbf{x}[p] = \mathbf{A}^{-1}\mathbf{y}[p]$$

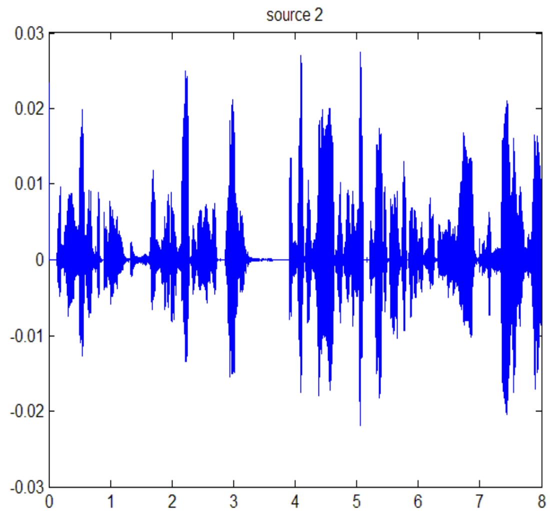
- Blind SIMO identification stage to recover primary sources from secondary ones,

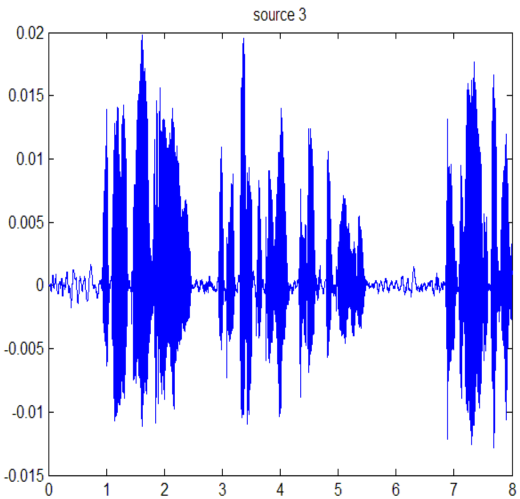
see, e.g., E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue, “Subspace Methods for the Blind Identification of Multichannel FIR filters,” *IEEE Trans. Signal Proc.*, vol. 43, pp. 516–525, 1995.

Practical example: separation of speech signals

- R=3 primary sources.
- Number of filters for each source: $L_1=L_2=L_3=4$; each filter of length 50 generated randomly → N=12 secondary sources.
- M=14 sensors (**A** is 14 by 12).
- K=21 covariance matrices.

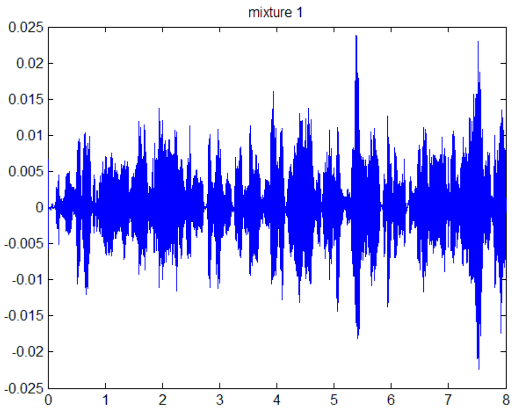




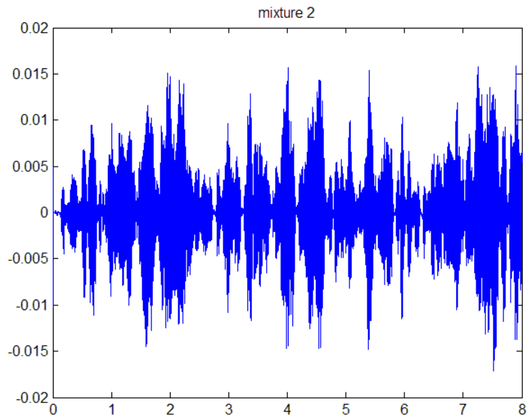


s_1 s_2 s_3

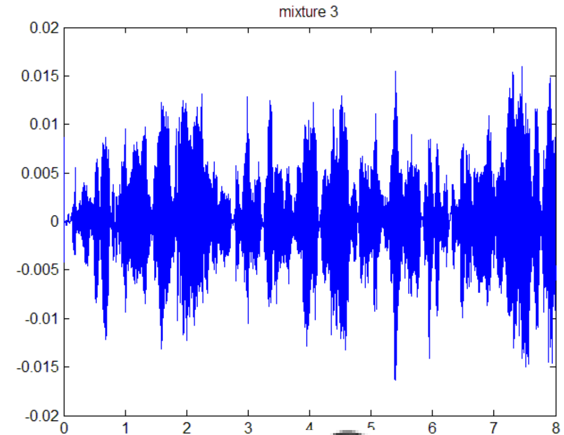
Practical example: separation of speech signals



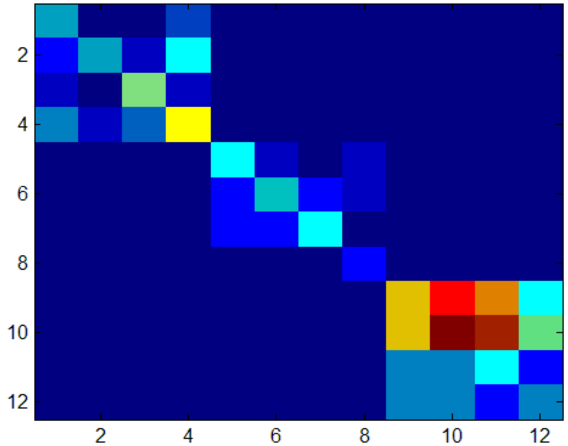
$\mathbf{y}_1$



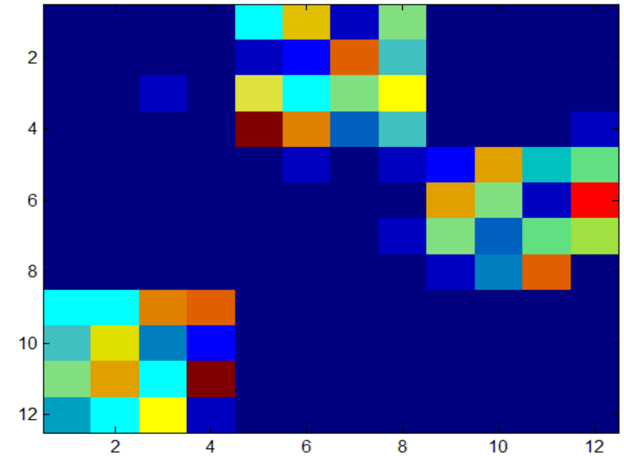
$\mathbf{y}_2$



$\mathbf{y}_3$ ... $\mathbf{y}_{14}$

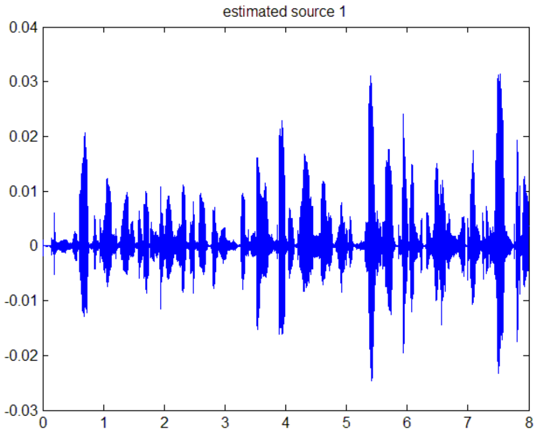


$$\sum_{k=1}^{K} \left| \mathbf{R}_{\mathbf{xx}}[\tau_k] \right| = \sum_{k=1}^{K} \left| \mathbf{D}_k \right|$$

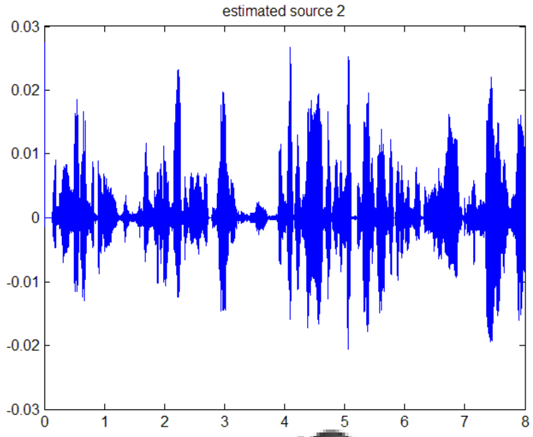


$$\mathbf{A}^{-1}\ \hat{\mathbf{A}}$$

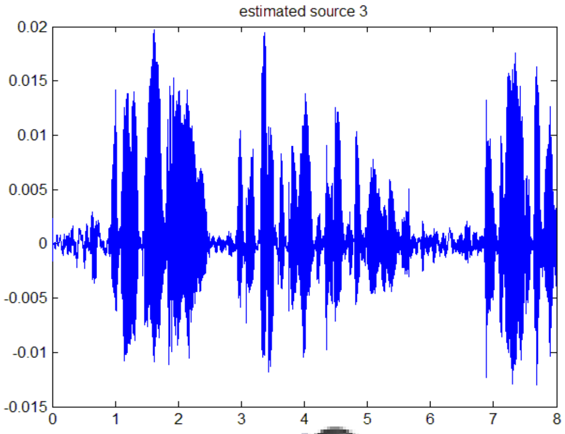
Practical example: separation of speech signals



$\hat{s}_1$



$\hat{s}_2$



$\hat{s}_3$

⇒ Separation is successful

Conclusion

- JBD is a generalization of JD
- JD ⟺ a particular case of Candecomp/Parafac

  JBD ⟺ a particular case of Block-Component-Decomposition (BCD)
- Uniqueness conditions for BCD can be invoked.
- In the exactly- and over- determined cases, we proposed an EVD-based technique useful for good initialization of JBD algorithms.
- We proposed a JBD-CG algorithm that works in exactly-, over- and under-determined cases
- Application: CG can accuratly achieve blind subspace separation based on Second Order Statistics.